

## 第一章

### 一. 名词解释

1. 操作系统: 集成软件和硬件资源, 管理软硬件的软件系统
4. 实时操作系统: 操作系统能及时响应外部事件的请求, 在规定的时间内完成对该事件的处理, 并控制所有实时任务协调一致地运行
5. 互斥共享: 多个进程或线程在访问共享资源时, 必须以互斥的方式进行访问

### 三. 填空

1. 硬件系统 软件系统
2. 硬件资源 软件资源
3. 共享性、异步性

## 第二章

### 一. 名词解释

3. 临界区: 每个进程中访问临界资源那段代码
4. 进程同步: 异步环境下的一组并发进程因互相制约而互相发送消息、互相合作等待, 使得各进程按一定的速度执行的进程

### 三. 填空

1. 动态性、并发性、独立性、异步性
5. 动态 静态
6. 互斥
8. 共享存储器 管道通信 消息传递
9. 就绪 执行 阻塞

### 五. 问答题

2. semaphore  $a=0, b=0, c=0, d=0, e=0, f=0, g=0, h=0$

<pre>P1() {     S1;     V(a);     V(b); }</pre>	<pre>P2() {     P(a);     S2;     V(c);     V(d); }</pre>	<pre>P3() {     P(b);     S3;     V(e); }</pre>	<pre>P4() {     P(c);     S4;     V(f); }</pre>	<pre>P5() {     P(d);     S5;     V(g); }</pre>	<pre>P6() {     P(e);     S6;     V(h); }</pre>	<pre>P7() {     P(f);     P(g);     P(h);     S7; }</pre>
---	---	---	---	---	---	---

3. semaphore  $empty=5, apple=0, orange=0, mutex=1$

<pre>father() {     while(1) {         P(empty);         P(mutex);         取苹果或桔子;         V(mutex);         if(取苹果) V(apple);         if(取桔子) V(orange);     } }</pre>	<pre>son() {     while(1) {         P(orange);         P(mutex);         取桔子;         V(mutex);         V(empty);     } }</pre>	<pre>daughter() {     while(1) {         P(apple);         P(mutex);         取苹果;         V(mutex);         V(empty);     } }</pre>
---	---	---



4. semaphore empty1=1, empty2=1, full1=0, full2=0

```
PA() {
    while(1) {
        P(empty1);
        将一个记录从磁盘读入缓冲区1;
        V(full1);
    }
}
```

```
PB() {
    while(1) {
        P(full1);
        复制缓冲区1的记录;
        V(empty1);
        P(empty2);
        复制的记录存入缓冲区2;
        V(full2);
    }
}
```

```
PC() {
    while(1) {
        P(full2);
        打印一个记录;
        V(empty2);
    }
}
```

5. ~~semaphore sofa=N, chair=1~~

semaphore customers=0, barber=0; mutex=1; pay=0

int waiting=0, chairs=N;

```
customers() {
    P(mutex);
    if(waiting < chairs) {
        waiting++;
        V(customers);
        V(mutex);
        P(barber);
        得到理发;
        支付离开;
        V(pay);
    } else {
        V(mutex);
        离开;
    }
}
```

```
barber() {
    while(1) {
        P(customers);
        P(mutex);
        waiting--;
        V(barber);
        V(mutex);
        理发;
        P(pay);
        P(mutex);
        V(barber);
        V(mutex);
    }
}
```

## 第三章

### 一、名词解释

2. 处理机调度：对处理机进行分配，按照一定算法决定哪些作业调入内存，哪些进程应获得处理机，哪些进程应调至外存等待

3. 周转时间：从作业被提交给系统开始到作业完成为止的时间间隔

4. 死锁：多个进程因竞争系统资源，相互等待对方占有的资源，导致进程永远处于阻塞状态，无法运行

### 三、填空

2. 提交、后备、运行、等待

3. 预防死锁、避免死锁、检测死锁、解除死锁



### 1. ① 按序法服务 FCFS

	完成时间	周转时间	带权周转时间
A	3	3	1
B	9	7	$7/6 = 1.167$
C	13	9	$9/4 = 2.25$
D	18	12	$12/5 = 2.4$
E	20	12	$12/2 = 6$

平均周转时间 = 8.6

平均带权周转时间 = 2.2634

### ② 按长SJF

	完成时间	周转时间	带权周转时间
A	3	3	$3/3 = 1$
B	15	13	$13/6 = 2.17$
C	8	4	$4/4 = 1$
D	20	14	$14/5 = 2.8$
E	10	2	$2/2 = 1$

平均周转时间 = 7.2

平均带权周转时间 = 1.594

### ② 按短SJF

	完成时间	周转时间	带权周转时间
A	3	3	$3/3 = 1$
B	9	7	$7/6 = 1.17$
C	15	11	$11/4 = 2.75$
D	20	14	$14/5 = 2.8$
E	11	3	$3/2 = 1.5$

平均周转时间 = 7.6

平均带权周转时间 = 1.844

### ④ 高响应比优先 HRRN

	完成时间	周转时间	带权周转时间
A	3	3	$3/3 = 1$
B	9	7	$7/6 = 1.17$
C	13	9	$9/4 = 2.25$
D	20	14	$14/5 = 2.8$
E	15	7	$7/2 = 3.5$

平均周转时间 = 8

平均带权周转时间 = 2.144

### ⑤ 时间片轮转

	完成时间	周转时间	带权周转时间
A	3	3	$3/3 = 1$
B	18	16	$16/6 = 2.67$
C	15	11	$11/4 = 2.75$
D	20	14	$14/5 = 2.8$
E	16	8	$8/2 = 4$

平均周转时间 = 10.4

平均带权周转时间 = 2.644

2. 0ms时: A的松弛度 =  $20 - 10 = 10ms$

A2 =  $40 - 10 = 30$  B的松弛度 =  $50 - 10 = 40ms$

C的松弛度 =  $50 - 15 = 35ms$

故执行A

10ms: A1执行完毕 A2的松弛度 =  $40 - 10 - 10 = 20$

B的松弛度 =  $50 - 10 - 10 = 30$

C的松弛度 =  $50 - 10 - 15 = 25$

故执行A2

20ms: A2执行完毕, 执行

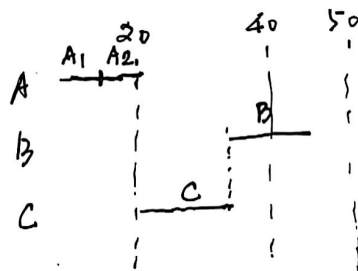
B的松弛度 =  $50 - 20 - 10 = 20$

C的松弛度 =  $50 - 20 - 15 = 15$

故执行C

35ms: C执行完毕, 执行B

45ms: B执行完毕



3. (1) 进程	Allocation
P0	0 0 3 2
P1	1 0 0 0
P2	1 3 5 4
P3	0 3 3 2
P4	0 0 1 4

该状态安全  $P0 \rightarrow P3 \rightarrow P4 \rightarrow P1 \rightarrow P2$

(2) 不能, 分配之后可利用资源向量为 0 4 0 0, 不能满足任何进程

(3) 不会立即死锁, 外面还剩 4 个资源可以分配

